GUNADARMA UNIVERSITY

MASTER DEGREE PROGRAM

A/B Testing Mekargo.id Website of Mekar PT Sampoerna Wirausaha with Bayesian Inference and Pymc3

Written by:

Name	**:**	**Alvian Dwi Kurnianto, S.Kom.**
Student Id	**:**	**92216007**
Supervisor	**:**	**Dr. Setia Wirawan**

Submitted as a Partial Fulllment of the Requirements for

Magister Degree of Information System Management

Concentrating on Information System Software

Depok

2018

PAGE OF APPROVAL

Title	:	**"A/B Testing Mekargo.id Website of Mekar PT Sampoerna Wirausaha with Bayesian Inference and Pymc3"**
Name	:	**Alvian Dwi Kurnianto, S.Kom.**
NIM	:	**92216007**
Date of Graduation	:	May, 12 2017

Approved by:

Board of Advisors

....

Dr. Setia Wirawan

(Chair Person)

....

Prof. Dr. Yuhara Sukra., M.Sc.

(Member)

....

Dr. Tb. Maulana Kusuma, S.Kom., MEngSc.

(Director)

ABSTRACT

Alvian Dwi Kurnianto, S.Kom.. 92216007
"A/B TESTING MEKARGO.ID WEBSITE OF MEKAR PT SAMPOERNA WIRAUSAHA WITH BAYESIAN INFERENCE AND PYMC3".
Master Thesis, Information System Management, Information System Software, Gunadarma University, 2018.
Keyword : A/B Testing, Bayesian Inference, Pymc3, Website, Google Analytics.

(xiii+ 54)

References (2013-2018)

ACKNOWLEDGEMENTS

Praise and thanks raised to Allah SWT for all blessing and also utterance and good wishes sent to the prophet of Muhammad SAW so, with all the grants, author have finished this thesis, which is titled “A/B Testing Mekargo.id Website of Mekar PT Sampoerna Wirausaha with Bayesian Inference and Pymc3”. This undergraduate thesis is intended to complete the requirement to finish my study in the Informatics Engineering Department, Gunadarma University. This writing would not have been possible without the support and encouragement from people around me. Therefore, I would like to say thank to:

.

Depok, February 2018

Alvian Dwi Kurnianto, S.Kom.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 Background

Usaha Kecil Menengah (UKM) memiliki peran penting dalam mendorong pertumbuhan perekonomian Indonesia. Dengan adanya sektor UKM, pengangguran akibat angkatan kerja yang tidak terserap dalam dunia kerja menjadi berkurang. Sektor UKM pun telah terbukti menjadi pilar perekonomian yang tangguh. Kontribusi sektor UKM dalam menentukan Produk Domestik Bruto (PDB) dan sektor penghasil devisa negara juga tak perlu diragukan lagi. Saat ini, UKM telah dijadikan agenda utama pembangunan ekonomi Indonesia (Kemenkeu, 2015).

Masalah mendasar UKM yang paling menonjol menyangkut menyediakan pembiayaan usaha alias modal usaha. Kebutuhan modal sangat terasa pada saat seseorang ingin memulai usaha baru. Alhasil, biasanya bila motivasinya kuat, seseorang akan tetap memulai usaha dengan modal seadanya. Pada usaha yang sudah berjalan, modal tetap menjadi kendala lanjutan untuk berkembang. Masalah yang menghadang UKM menyangkut kemampuan akses pembiayaan, akses pasar dan pemasaran, tata kelola manajemen usaha kecil serta akses informasi. Kesulitan UKM mengakses sumber-sumber modal karena keterbatasan informasi dan kemampuan menembus sumber modal tersebut. Padahal pilihan sumber modal sangat banyak dan beragam. Lembaga keuangan bank adalah sumber modal terbesar yang dapat dimanfaatkan oleh pelaku usaha kecil. Namun untuk bermitra dengan bank, usaha kecil dituntut menyajikan proposal usaha yang feasible atau layak usaha dan menguntungkan (Wuisan, 2017).

Mekar (PT Sampoerna Wirausaha) didirikan pada 2010 dan dimiliki secara penuh oleh Putera Sampoerna Foundation, Mekar bertujuan untuk mempermudah akses finansial bagi para pelaku UKM Indonesia sehingga memiliki dampak sosial dan ekonomi yang positif di Indonesia (Mekar, 2017). Mekar memberikan solusi dalam membantu Pelaku Usaha Mikro Kecil dan Menengah (UMKM) dan Konsumen lainnya untuk mendapatkan Akses Layanan Keuangan atau Permodalan di Indonesia. Beberapa produk yang dimiliki Mekar diantaranya website utama Mekar.id, chat bot dan website Mekar Go. Website Mekar Go dengan alamat web https://mekargo.id/ merupakan website untuk menghimpun data UKM dan karyawan yang ingin mengajukan pinjaman ke Mekar. Data yang telah dihimpun pada versi web Mekar GO saat ini menunjukkan konversi peminjam yang melengkapi pengisian data dari awal sampai akhir kurang dari 10%. Presentase konversi ini menurut tim marketing terlalu kecil dan perlu ditingkatkan.

Untuk meningkatkan presentase konversi pelengkapan data, tim marketing mengusulkan perubahan urutan pengisian data. Perubahan urutan pengisian data dibuat menjadi dua versi baru yang selanjutnya disebut sebagai versi B dan versi C, dan versi saat ini disebut sebagai versi A. Ketiga versi tersebut akan dilakukan A/B testing untuk mengetahui versi manakah yang menghasilkan presentasi konversi pelengkapan data terbanyak. A/B testing adalah proses yang digunakan dalam marketing untuk mengisolasi dan menguji faktor-faktor yang mempengaruhi performa versi produk mana yang memenuhi kriteria marketing (Dixon et al., 2015). A/B testing banyak digunakan untuk pengoptimalan situs web: dua versi halaman web, katakanlah A dan B, secara empiris dibandingkan dengan disajikan kepada pengguna. Setiap pengguna hanya melihat salah satu dari dua versi, dan tujuannya adalah untuk menentukan versi mana yang lebih baik. Tujuan yang ingin dicapai pada umumnya adalah untuk menentukan halaman web mana yang memiliki tingkat konversi tertinggi (kemungkinan pengguna benar-benar menjadi pelanggan) dengan menerima umpan balik dari pengguna (Kaufmann et al., 2014). Dalam penelitian ini, pengumpulan data umpan balik dari pengguna dibantu dengan tools Google Analytics. Google Analytics merupakan analytics tool yang dibuat oleh Google, digunakan untuk mengukur performa website dan perilaku pengguna saat mengunjungi website (Yang and Perrin, 2014). Setelah data yang dikumpulkan oleh Google Analytics terpenuhi, dilakukan proses analisis data dengan metode bayessian inference dan tools pymc3. Pymc3 adalah framework probabilistik programming dengan bayessian inference yang ditulis dalam bahasa pemrograman python (Salvatier et al., 2016).

1.2 Problem Identification

Adapun masalah yang dibahas pada penelitian ini diantaranya adalah bagaimana metode yang dilakukan untuk melakukan A/B testing website mekargo.id pada PT Mekar Sampoerna Wirausaha, bagaimana cara menganalis hasil A/B testing dengan metode bayesian inference dengan tools pymc3, dan versi manakah dari A/B testing yang menjadi versi terbaik.

1.3 Scope of Problem

Adapun batasan masalah dari penelitian ini adalah sebagai berikut:

1. Objek penelitian adalah website mekargo.id.
2. Penelitian dilakukan di PT Mekar Sampoerna Wirausaha.
3. Waktu penelitian antara Agustus 2017 sampai Januari 2018.
4. Penelitian menerapkan taknik A/B testing pada website mekargo.id dengan bantuan tools Google Analytics.
5. Data yang dihasilkan dari A/B testing dilakukan analisis dengan metode bayesian inference menggunakan tools library pymc3, numpy, matplotlib, scipy, jupyter dan Ipython dengan bahasa pemrograman python.
6. Fitur website mekargo.id yang dibahas adalah fitur pengisian data peminjam.
7. Fitur website mekargo.id yang tidak dibahas adalah fitur static page yang berisi halaman home, about us, terms and conditions, privacy policy dan contact us.

1.4 Research Purposes

Tujuan dari penelitian ini adalah melakukan A/B testing web mekargo.id pada PT Mekar Sampoerna Wirausaha, melakukan analisis hasil A/B testing dengan metode bayesian inference dengan pymc3, mengetahui versi pada A/B testing yang menjadi versi terbaik.

1.5 Research Benefits

Manfaat yang didapat dari penelitian ini adalah:

1. Bagi peneliti, menjadi panduan atau referensi penelitian di bidang sistem informasi pada penelitian selanjutnya.
2. Bagi peneliti lain, sebagai sumber literatur untuk penelitian yang sejenis.
3. Bagi perusahaan yang diteliti, memberikan saran perbaikan dengan terpilihnya versi A/B testing yang terbaik.

Chapter 2

LITERATURE REVIEW

2.1 Python

Python is a dynamically typed programming language that has a focus on ease of use and readability. Due in part to this focus, it has become a popular language for scientific computing and data science, with a broad ecosystem of libraries (Meurer et al., 2017). Python syntax is designed very readable, which is important for software development. Every computer program is written only once, but read and revised many times, often by many people. Being readable also makes it easier to learn and remember, hence more writeable. Compared with other popular languages, Python has a gentle learning curve that makes developer more productive, yet it has depths that can be explored and gain expertise (Lubanovic, 2014).

People use python because have these several advantages: software quality, developer productivity, program portability, support libraries, component integration, and enjoyment (Lutz, 2013). For application development, Python has advadtages of ease for development and prototyping (Vohl et al., 2016). Python is a multi-platform, general-purpose programming language that can run on Windows, Linux/Unix, and Mac OS X, and has been ported to Java and .NET virtual machines as well. It has a powerful standard library. In addition, it has many libraries for data analysis such as Numpy, Pandas, Matplotlib, PyMongo, and Scikit (Vo et al., 2015). Python is the most popular programming languages in the scientific domains (Rougier et al., 2017). The frequency of programming languages appearing in Journal of Open Source Software (JOSS) articles show that Python appears the most with over half of published articles in total 54 (Smith et al., 2018). The statistic of Python programming language that appeared in JOSS is shown by figure 2.1.

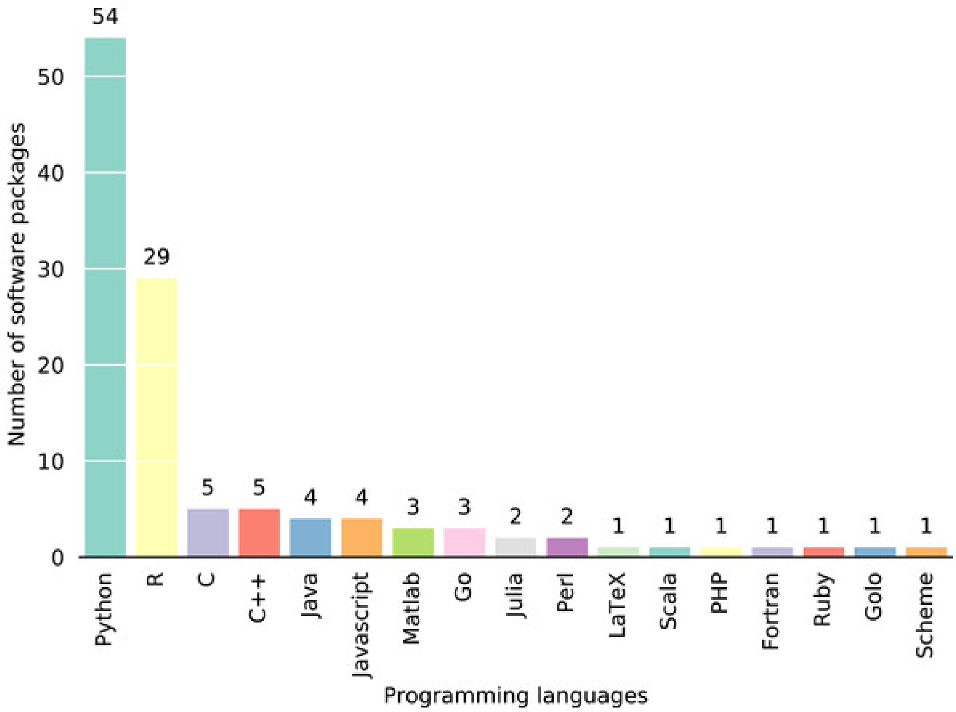


Figure 2.1: Python programming language usage in JOSS articles
Source: Smith et al., 2018

2.2 Numpy

Numpy is a fundamental package that is extremely useful for scientific computing, and contains among other things a powerful N-dimensional array object (Strickland et al., 2014). The Numpy package, which comprises the Numpy array as well as a set of accompanying mathematical functions, has found wide-spread adoption in academia, national laboratories, and industry, with applications ranging from gaming to space exploration (Van Der Walt et al., 2016).

2.3 Matplotlib

Matplotlib is a python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in python scripts, the python and ipython shell, web application servers, and six graphical user interface toolkits (Hunter et al., 2014). Matplotlib provides both a very quick way to visualize data from Python and publication-quality figures in many formats: line plots, contour plots, scatter plots, and basemap plots. It comes with a set of default settings, but allows customization of all kinds of properties. However, we can easily create our chart with the defaults of almost every property in Matplotlib (Vo et al., 2015). Some of charts example that can be generated by matplotlib are ilustrated by figure 2.2.

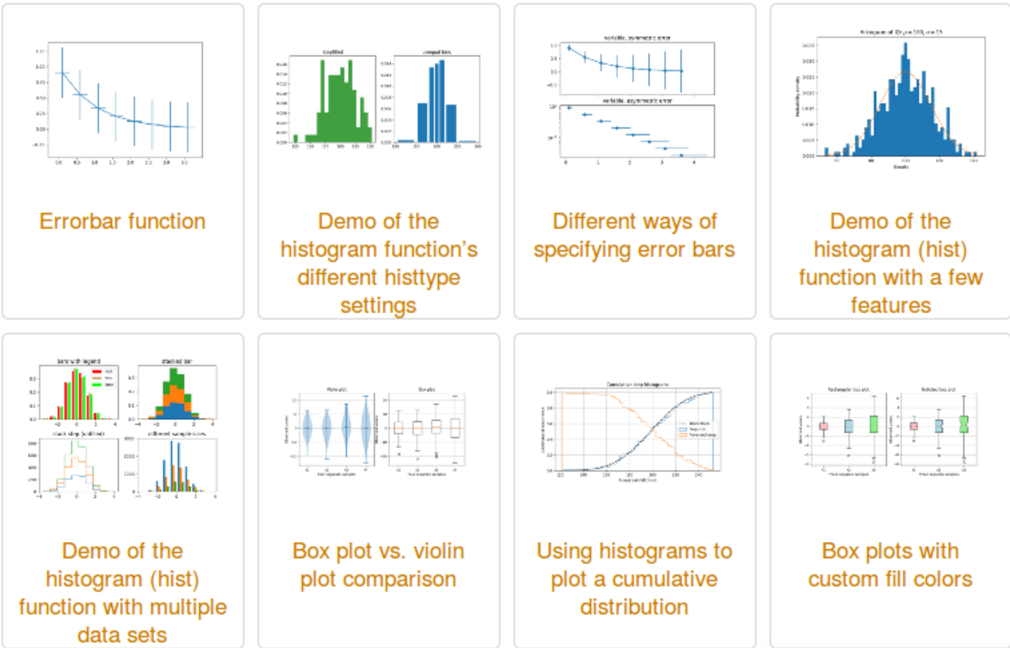


Figure 2.2: Some examples of matplotlib chart
Source: Hunter et al., 2014

2.4 Scipy

Scipy (pronounced "Sigh Pie") is a Python-based ecosystem of open-source software for mathematics, science, and engineering. Scipy ecosystem includes general and specialised tools for data management and computation, productive experimentation and high-performance computing. The SciPy library, a collection of numerical algorithms and domain-specific toolboxes, including signal processing, optimization, statistics and much more (Jones et al., 2014).

2.5 Ipython

Ipython is an interactive browser-based environment where you can combine code execution, text, mathematics, plots, and rich media into a single document. Originally designed for use as an electronic lab notebook for computational science, it is increasingly being used in teaching as well, and a rich ecosystem of open source plugins and extensions for teaching is growing around it (Wilson et al., 2014). IPython has provided terminal-based tools for interactive computing in Python since 2001. The notebook document format and multi-process architecture introduced in 2011 have expanded the applicable scope of IPython into teaching, presenting, and sharing computational work, in addition to interactive exploration (Ragan-Kelley et al., 2014).

2.6 Jupyter

Jupyter Notebook, provides a tool to create and share web pages with text, charts, and Python code in a special format. Often, the notebooks are used as an educational tool, or to demonstrate Python software. We can import or export notebooks either from plain Python code or from the special notebook format. The notebooks can be run locally, or we can make them available online by running a dedicated notebook server (Fandango and Idris, 2017).

Jupyter Notebook is accessed through a modern web browser. This makes it practical to use the same interface running locally like a desktop application, or running on a remote server. In the latter case, the only software the user needs locally is a web browser; so, for instance, a teacher can set up the software on a server and easily give students access. The notebook files it creates are a simple, documented JSON format, with the extension '.ipynb'. It is simple to write other software tools which access and manipulate these files (Kluyver et al., 2016). Jupyter notebook interface is shown by figure 2.3.

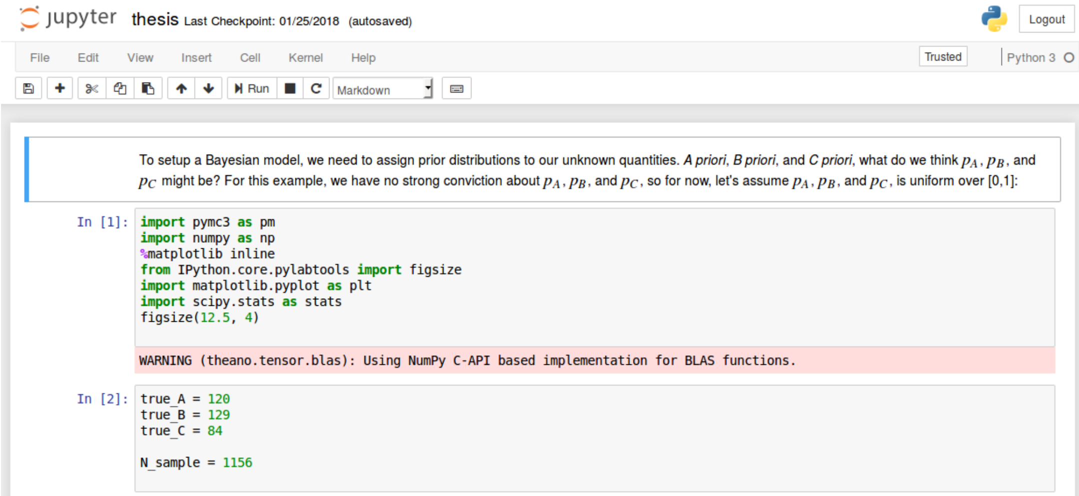


Figure 2.3: Jupyter notebook interface

2.7 Bayesian Inference

Bayesian inference is the process of fitting a probability model to a set of data and summarizing the result by a probability distribution on the parameters of the model and on unobserved quantities such as predictions for new observations (Gelman et al., 2014). Bayesian inference differs from more traditional statistical inference by preserving uncertainty. The Bayesian worldview interprets probability as measure of believability in an event, that is, how confident we are in an event occurring. In fact, we will see in a moment that this is the natural interpretation of probability (Davidson-Pilon, 2015).

Bayesian inference departs from the approach to statistical inference described in many textbooks, which is based on a retrospective evaluation of the procedure used to estimate θ (or $\tilde{y}$) over the distribution of possible y values conditional on the true unknown value of θ. Despite this difference, it will be seen that in many simple analyses, superficially similar conclusions result from the two approaches to statistical inference. However, analyses obtained using Bayesian methods can be easily extended to more complex problems (Gelman et al., 2014).

The Bayesian modeling paradigm involves setting up a probabilistic model describing how data was gen- erated, assigning prior distributions over unknown model parameters, and then calculating a posterior distribution over these parameters. In practice, this posterior distribution is often intractable to compute exactly except for the simplest models. This often requires one to resort to approximate posterior inference algorithms based on Monte-Carlo sampling or Variational Inference (VI). Fortunately, a lot of progress has been made recently in this area including state-of-the-art algorithms such as Hamiltonian Monte Carlo sampling (HMC), the No-U-Turn Sampler (NUTS), Automatic Differentiation VI (ADVI) and Black Box VI. These algorithms are applica- ble to a wide class of models and are readily available in popular Probabilistic Programming languages such as Stan, Edward, and PyMC3 (Pourzanjani et al., 2017). Bayesian inference had been applied in many areas such as automatic machine-learning hyperparameter optimization, A/B testing or recommender systems, among others (Jiménez and Ginebra, 2017).

Some advantages of the Bayesian approach are: that it provides a transparent framework for inference; secondly, it is flexible and often provides an elegant and practical solution to inference arising from very complex statistical models. Note, in particular, that Bayesian methods make no formal distinction between estimation oand prediction, and in this way naturally incorporate parameter uncertainty into predictive inference. Because, obtaining the sample can itself be a major challenge (Taylor et al., 2015).

2.8 Markov Chain Monte Carlo

Markov Chain Monte Carlo (MCMC) method is used to generate samples from a probability distribution. It is widely used nowadays in many aspects of optimization and numerical integration, and is specially suitable for being used in sampling from posterior distributions in Bayesian inference. A key issue in MCMC is whether the chain has converged and is actually sampling from the target distribution. There is not a single infallible test of convergence, but many formal and informal ways to assess non-convergence (Fernández-i Marín et al., 2016).

Markov chain Monte Carlo is a general method based on drawing values of θ from approximate distributions and then correcting those draws to better approximate the target posterior distribution, $p(\theta|y)$. The sampling is done sequentially, with the distribution of the sampled draws depending on the last value drawn; hence, the draws form a Markov chain. (As defined in probability theory, a Markov chain is a sequence of random variables θ^1 , θ^2 , . . ., for which, for any t, the distribution of θ^t given all previous θ's depends only on the most recent value, θ^{t-1} .) The key to the method's success, however, is not the Markov property but rather that the approximate distributions are improved at each step in the simulation, in the sense of converging to the target distribution (Gelman et al., 2014).

2.9 Pymc3

PyMC3 is a new, open-source probabilistic programming framework with an intuitive and readable, yet powerful, syntax that is close to the natural syntax statisticians use to describe models. It features next-generation Markov chain Monte Carlo (MCMC) sampling algorithms (Salvatier et al., 2016). Running PyMC3 requires a working Python interpreter (Python Software Foundation, 2010), either version 2.7 (or more recent) or 3.4 (or more recent). PyMC3 can be installed using 'pip': pip install pymc3. PyMC3 depends on several third-party Python packages which will be automatically installed when installing via pip (Salvatier et al., 2016).

The majority of the heavy lifting done by PyMC3 is taken care of with the theano package. The notation in theano is remarkably similar to NumPy. It also supports many of the familiar computational elements of NumPy. However, while NumPy directly executes computations, e.g. when you run a + b, theano instead builds up a "compute graph" that tracks that you want to perform the + operation on the elements a and b. Only when you eval() a theano expression does the computation take place (i.e. theano is lazy evaluated). Once the compute graph is built, we can perform all kinds of mathematical optimizations (e.g. simplifications), compute gradients via autodiff, compile the entire graph to C to run at machine speed, and also compile it to run on the GPU. PyMC3 is basically a collection of theano symbolic expressions for various probability distributions that are combined to one big compute graph making up the whole model log probability, and a collection of inference algorithms that use that graph to compute probabilities and gradients (Davidson-Pilon, 2015).

2.10 Theano

Theano is a Python library that allows to define, optimize, and evaluate mathematical expressions involving multi-dimensional arrays efficiently. Since its introduction in it has been one of the most used CPU and GPU mathematical compilers – especially in the machine learning community and has shown steady performance improvements. Theano is being actively and continuously developed since 2008, multiple frameworks have been built on top of it and it has been used to produce many state-of-the-art machine learning models. Theano allows a user to symbolically define mathematical expressions and have them compiled in a highly optimized fashion either on CPUs or GPUs (the latter using CUDA) 1 , just by modifying a configuration flag. Furthermore, Theano can automatically compute symbolic differentiation of complex expressions, ignore the variables that are not required to compute the final output, reuse partial results to avoid redundant computations, apply mathematical simplifications, compute operations in place when possible to minimize the memory usage, and apply numerical stability optimization to overcome or minimize the error due to hardware approximations. To achieve this, the mathematical expressions defined by the user are stored as a graph of variables and operations, that is pruned and optimized at compilation time.

The interface to Theano is Python, a powerful and flexible language that allows for rapid prototyping and provides a fast and easy way to interact with the data. The downside of Python is its interpreter, that is in many cases a poor engine for executing mathematical calculations both in terms of memory usage and speed. Theano overcomes this limitation, by exploiting the compactness and ductility of the Python language and combining them with a fast and optimized computation engine. Theano's API mimics NumPy, a widely adopted Python library that provides an n-dimensional array data type and many functions for indexing, reshaping, and performing elementary computations (exp, log, sin, etc.) on entire arrays at once. This allows Python users to rapidly switch to Theano using a familiar syntax and set of instructions – extended with advanced features, such as automatic gradient computation, numerical stability improvements and optimization – and generate a high-performance code for CPU as well as for GPU, without requiring changes to the user code. Theano has also been designed for easy and fast extensibility through the definition of custom graph expressions written in Python, C++, or CUDA (Al-Rfou et al., 2016).

2.11 Google Analytics

Google Analytics (GA) is an easy-to-use tool to measure activity on a website. A basic setup might take as little as a few minutes, and many of the standard reports are quite accessible and understandable without any special training or prior knowledge of web analytics. Because of this, many users jump into GA without knowing much about its underpinnings—how the data is structured and gathered—and that's fine for the basics. But eventually, users can outgrow this intuitive understanding of GA and its data, and need deeper insight into how it works and what it can do (Weber, 2015).

Google Analytics is free, and it is always increasing in power, to rival and in some cases exceed the performance of the "paid" tools (Kelsey, 2017). Google Analytics has another advantages such as has tremendous features, and ease of use. Unlike other Web analytics tools that use server log files, Google collects information by inserting simple Javascript codes into Web pages. The advantage of this method is that Google Analytics can capture technical and demographic information that log files do not normally provide, such as the user's browser, operating system, screen size, resolution, and so on (Yang and Perrin, 2014). The example of Google Analytics dashboard is shown by figure 2.4.

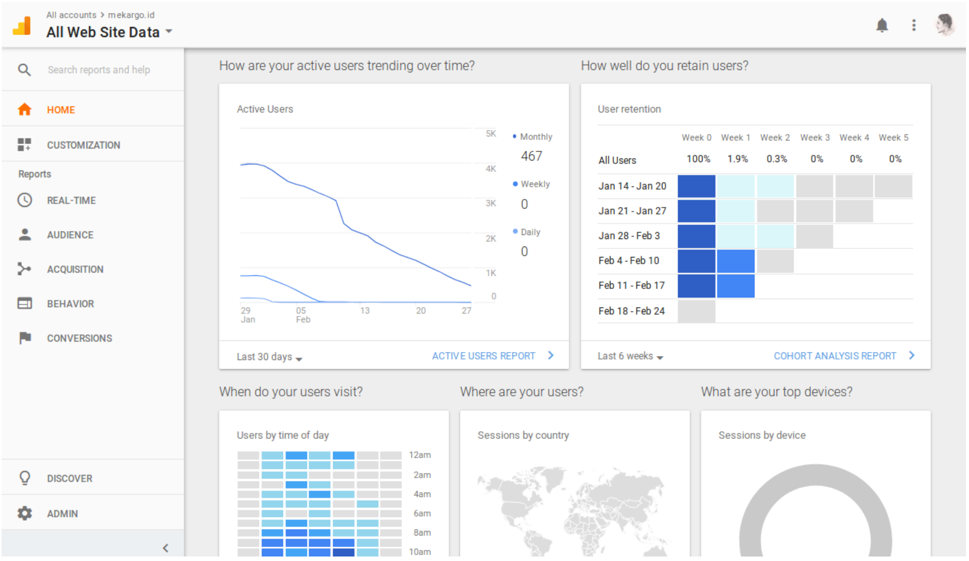


Figure 2.4: Google Analytics dashboard

2.12 A/B Testing

A/B testing has widely become the standard controlled experimentation framework for network driven companies like Facebook, LinkedIn, Twitter, Google, and Yahoo (Wilson and Uminsky, 2017). A/B testing is ubiquitous within the machine learning and data science operations of internet companies. Generically, the idea is to perform a statistical test of the hypothesis that a new feature is better than the existing platform. For example, it results in higher revenue (Goldberg and Johndrow, 2017).

A/B testing started to be used in the late 1990s with the growth of the Internet. Many large companies run thousands to tens of thousands of A/B testing each year testing user interface (UI) changes, enhancements to algorithms (search, ads, personalization, recommendation, etc.), changes to apps, content management system, etc. In an A/B testing, users are randomly split between the variants (e.g., the two different ads layouts) in a persistent manner (a user receives the same experience in multiple visits). Their interactions with the site are instrumented and key metrics computed (Kohavi and Longbotham, 2017).

In A/B testing, one has a proposed new version of a software platform and wants to decide whether or not to ship the new version. The classical way of conceiving of this problem is the following. We divide users into two groups: treatment and control. We then roll out the proposed update to the treatment group while leaving the control group with the current version. Using data gathered from this randomized trial, we then ask whether the new version performed “better” with respect to some metric (Goldberg and Johndrow, 2017). A/B testing can be ilustrated by figure 2.5.

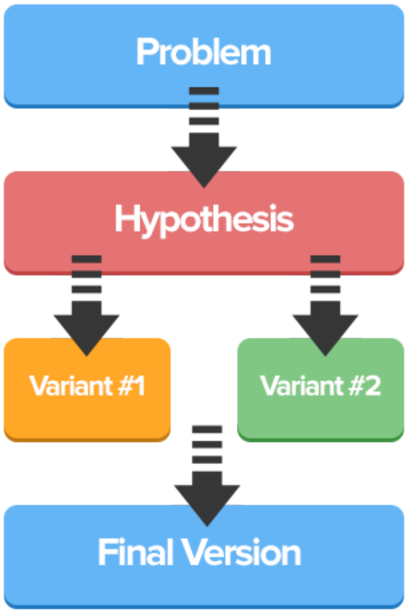


Figure 2.5: A/B Testing ilustration
Source: Omniconvert, 2017

2.13 Previous Researches

Table 2.1 contains previous researches on A/B testing. It contains number, title of the article, authors of the article, and strength and of the article from author point of view. There are six articles that author discuss. The point of view is composed from the mathematical concept explanation, application example, and suplementary material of the articles.

Table 2.1: Previous research

No	Article	Authors	Strength	Weakness

No	Article	Authors	Strength	Weakness
1	A Decision Theoretic Approach to A/B Testing(Goldberg and Johndrow, 2017)	David Goldberg (eBay) and James E. Johndrow (Stanford University)	Great on Mathematical concept explanation	Lack of application example
2	Always Valid Inference: Bringing Sequential Analysis to A/B Testing(Johari et al., 2015)	Ramesh Johari, Leo Pekelis and David J. Walsh	Great on real world example and Mathematical concept explanation. Also contains Supplementary Material	There is no proof of theorem explanation
3	Online Controlled Experiments and A/B Testing(Kohavi and Longbotham, 2017)	Ron Kohavi (Microsoft) and Roger Longbotham (Microsoft)	Great on real world example	Lack of Mathematical concept explanation

No	Article	Authors	Strength	Weakness
4	Online Controlled Experiments at Large Scale (Kohavi et al., 2013)	Ron Kohavi (Microsoft), Alex Deng (Microsoft), Brian Frasca (Microsoft), Toby Walker (Microsoft), Ya Xu (Microsoft), and Nils Pohlmann (Microsoft)	Great on real world example	Lack of Mathematical concept explanation
5	On the Complexity of A/B Testing (Kaufmann et al., 2014)	Emilie Kaufmann (LTCI, Télécom ParisTech & CNRS) , Olivier Cappé (LTCI, Télécom ParisTech & CNRS), and Aurélien Garivier (Institut de Mathématiques de Toulouse, Université Paul Sabatier)	Great on Mathematical concept explanation	Lack of application example

No	Article	Authors	Strength	Weakness
6	The Power of A/B Testing under Interference (Wilson and Uminsky, 2017)	James D. Wilson (Department of Mathematics and Statistics University of San Francisco) and David T. Uminsky (Department of Mathematics and Statistics University of San Francisco)	Great on real world example and Mathematical concept explanation. Also, completed with proof of theorem explanation	There is no Supplementary Material

Chapter 3

RESEARCH METHODOLOGY

3.1 Research Object

Waktu penelitian dilakukan antara bulan Agustus 2017 sampai Januari 2018. Objek pada peneltian ini adalah website mekargo.id milik Mekar. Mekar hadir melalui Mekar Go untuk memberikan solusi dalam membantu pelaku UMKM dan konsumen lainnya untuk mendapatkan akses layanan keuangan atau permodalan di Indonesia. Mekar memberikan kemudahan akses melalui portal Mekar Go kepada lembaga keuangan seperti Bank, perusahaan pembiayaan, dan Bank Perkreditan Rakyat dalam meningkatkan kualitas pelayanan kepada masyarakat secara mudah, aman dan berkualitas.

Fitur yang diteliti adalah fitur pendaftaran bagi peminjam. Data yang digunakan adalah alamat url yang dikunjungi pendaftar untuk masing-masing versi A/B testing. Pengumpulan data dilakukan dengan tools google analytics. Data yang didapatkan selanjutnya akan dianalisis dengan metode bayesian inference dan pymc3.

3.2 Research Methodology

Metode penelitian yang dilakukan meliputi: identifying problems, defining website measurement, developing a hypothesis, developing and testing page variants, dan analyzing test results. Gambaran langkah metode penelitian yang dilakukan dapat diilustrasikan oleh figure 3.1.

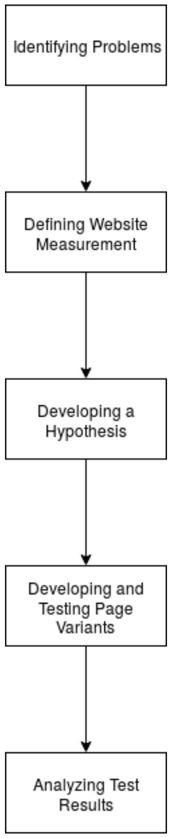


Figure 3.1: Research Methodology

3.2.1 Identifying Problems

Masalah yang diidentifikasi adalah yang berhubungan dengan website yang akan dilakukan A/B testing. Masalah yang diangkat biasanya berhubungan dengan bisnis, marketing, atau convertion rate. Tergantung pada tujuan apa yang ingin dicapai pada pembuatan A/B testing pada website tersebut. Masalah yang dijabarkan biasanya dikarenakan ada suatu hal yang timbul yang tidak sesuai tujuan. Selain itu, dijabarkan pula kondisi saat ini dari website tersebut.

3.2.2 Defining Website Measurement

Website measurement adalah pengukuran yang dilakukan untuk memberikan kejalasan untuk mencapai target A/B testing yang diinginkan. Website measurement terdiri dari beberapa poin indikator bisnis yang dijabarkan dari hal yang umum sampai hal yang detil. Poin indikator ini terdiri dari business objectives, website goals, key performance metrics (KPIs), dan target metrics. Dengan adanya penjabaran ini, akan lebih memperjelas apa hasil yang diinginkan dari A/B testing ini. Sehingga, seluruh anggota tim yang terlibat dapat menyatukan visi dalam melakukan A/B testing ini. Diagramnya dapat diilustrasikan pada figure 3.2.

Figure 3.2: Website measurement
Source: Omniconvert, 2017

Adapun penjabaran dari poin indikator website measurement tersebut adalah sebagai berikut:

- **Business objectives** adalah sasaran utama yang ingin dicapai pada pembuatan A/B testing. Misalnya meningkatkan jumlah penjualan, menambah jumlah user yang mendaftar, atau meningkatkan pengunjung pada tampilan halaman. Poin ini bergantung pada jenis bisnis yang dijalankan dan tahap apa yang sedang terdapat masalah.

- **Website goals** adalah tindakan apa yang ingin dicapai atau ditingkatkan pada satu fitur website yang dapat memenuhi sasaran poin business objective. Jika ingin meningkatkan jumlah penjualan, maka website goal yang harus dicapai adalah peningkatan fitur menambahkan barang ke chart sampai melakukan pembayaran. Website goal bergantung kepada poin business objective yang ingin dicapai.

- **Key performance metrics** adalah ukuran yang menjadi pedoman keberhasilan terhadap hal yang ingin dicapai. Misalnya ingin meningkatkan penjualan, maka ukuran yang dipakai adalah berapa jumlah barang yang dimasukkan ke dalam chart, berapa jumlah barang pada chart yang dilakukan checkout pembayaran, dan sebagainya.

- **Target metrics** adalah berapa target angka yang dijadikan target keberhasilan dari tiap key performance metric disertai dengan jangka waktu yang menjadi batasan. Misalnya, target jumlah barang yang dimasukkan ke dalam chart minimal 1.000.000 barang per hari. Itu adalah salah satu contoh untuk perusahaan ecommerce yang sudah bersar. Target angka yang dijadikan ukuran bergantung pada kondisi saat ini juga. Yang terpenting adalah adanya peningkatan.

3.2.3 Developing a Hypothesis

A hypothesis will define why you believe a problem occurs. Asumsi awal yang menjadi penyebab masalah pada point 3.3.1 disertai dengan pengajuan solusi yang akan diterapkan. Menyebutkan hasil akhir dari solusi yang diajukan disertai alasannya.

Many of the variables specifically require you to bring data to the table to prioritize your hypotheses.

- Is it addressing an issue discovered via user testing?
- Is it addressing an issue discovered via qualitative feedback (surveys, polls, interviews)?
- Is the hypothesis supported by mouse tracking heat maps or eye tracking?
- Is it addressing insights found via digital analytics?

Having weekly discussions on tests with these 4 questions asked from everyone will quickly make people stop relying on just opinions. There are also bounds placed on Ease of implementation by bracketing answers according to the estimated time. Ideally you'd have a test developer be part of prioritization discussions.

A good hypothesis:

1. Is testable – hypothesis is measurable, so that it can be used in testing has a goal of solving conversion problems – Split testing is done to solve specific conversion problems
2. Gains market insights – Besides increasing conversion rates, split testing will give information about clients.
3. A well-articulated hypothesis will let split testing results give information about customers.

After you have defined your problem and articulated your hypothesis, you can start to come up with specific split-testing variations. Clearly defining your hypothesis will help you come up with testing variations that give meaningful results (CXL, 2017).

3.2.3.1 How to Develop Hypothesis

ada 4 cara yaitu bla bla bla bla

Case Studies : Conversion rate optimization relies heavily on case studies to guide best practices and develop new hypothesis. While it's not always true, case studies often represent solutions that might work in a given situation. Since at the hypothesis stage, your job is to collect as many plausible solutions as you can, digging through case studies can be extremely useful. However, not every case study will actually yield actionable insights to your specific case. When evaluating a case study, consider these issues:

- **Relevancy**: Is the case study describing a problem that is relevant to your particular problem?
- **Audience overlap**: Is the case study about a website that has a similar audience to yours?
- **Accuracy**: Are the results from the case study accurate enough? Was the testing carried out to a high degree of confidence? Far too many times, you'll find companies promoting case studies that are not statistically accurate.
- **Recency**: Are the findings from the case study recent enough? Have new trends impacted these results? What worked for a brand in 2006 might not hold true in 2016.
- **Scope**: Plenty of case studies focus on "small wins" or micro-conversions, such as the CTR for a button. Such results might look good, but unless they improve your target metrics, they are effectively meaningless. Always consider the scope of the study before borrowing its learnings in your own tests.

Best Practices: Best practices are just that - a practice assumed to be correct by default in any non-specific case. For example, placing your navigation bar at the top of the page is a "best practice". Best practices spring from three things:

- **Convention**: Some design choices are made just because that's the way things have always been done. In most cases, it's better to follow conventions since your customers are already used to it. Breaking convention might cause confusion.

- **Theory**: Some design choices spring from theory. For example, psychological theories state that we are likely to equate strong social proof with trustworthiness. Hence, including reviews, testimonials, etc. has now become a "best practice".

- **Testing**: When usability and CRO tests yield the same insights consistently, they often become best practices. For instance, testing shows that descriptive CTAs (such as "Download eBook") regularly perform better than plain CTAs (such as "Submit"). This can now be said to be a "best practice".

Theory: Conversion rate optimization sits at the intersection of sales, design and psychology. As such, any hypothesis you come up with will likely involve theories from each of these fields. To give you an example, good UI/UX design relies heavily on Gestalt psychology (such as the Law of Proximity and the Law of Similarity). This also informs the decisions you make when designing a conversion-optimized page. A strong understanding of theory will help you create better hypotheses. You don't have to know all of the above, but a firm grasp of psychological and emotional triggers that lead to sales will greatly help you improve conversions.

Experience: All the case studies and theories in the world can't make up for past experience. This is why CRO is often best left to experts; they can bring insights to the table that non-practitioners wouldn't know. It's not unusual to find that the most successful hypothesis is usually the one that springs from experience, not theory or case studies. A CRO expert who has already dealt with a site similar to yours in the same niche will likely have tons of ideas for improving your conversion rates (Omniconvert, 2017).

3.2.4 Developing and Testing Page Variants

Developing test variants berisi perubahan field pada form dari versi yang sebelumnya. Selain itu, terdapat juga pendefinisian nama halaman disertai url yang merepresentasi halaman tersebut. Perbedaan versi A, B, dan C ditentukan dari perbedaan urutan url dalam pengisian data peminjam. Cara pembagian traffic untuk setiap versi agar jumlah pengunjung berimbang juga dijelaskan. Testing page variants berisi penjelasan implementasi Google Analytics pada seluruh halaman. Hasil implementasinya dilihat pada dashboard akun Google Analytics dan dilakukan ekstrakting data.

3.2.5 Analyzing Test Results

3.2.5.1 Bayessian Inference Algorithm

Let $X_1, ..., X_n$ be n observations sampled from a probability density $p(x|\theta)$. Write $p(x|\theta)$ if θ as a random variable and $p(x|\theta)$ represents the conditional probability density of X conditioned on θ. In contrast, write $p_\theta(x)$ if θ as a deterministic value.

Algoritma 3.1 Bayesian inference algorithm

1. Choose a probability density $\pi(\theta)$ — called the prior distribution — that expresses beliefs about a parameter θ.

2. Choose a statistical model $p(x|\theta)$ that reflects beliefs about x given θ.

3. After observing data $D_n = \{X_1, ..., X_n\}$, update beliefs and calculate the posterior distribution $p(\theta|D_n)$. The posterior distribution can be written as:

$$p(\theta|X_1, ..., X_n) = \frac{p(X_1, ..., X_n|\theta)\pi(\theta)}{p\left(X_1, ..., X_n\right)} = \frac{\mathcal{L}_n\left(\theta\right)\pi\left(\theta\right)}{c_n}\alpha\mathcal{L}_n\left(\theta\right)\pi\left(\theta\right)$$

where $\mathcal{L}_n\left(\theta\right) = \prod_{i=1}^{n} p\left(X_i|\theta\right)$ is the likelihood function and

$$c_n = p\left(X_1, ..., X_n\right) = \int p(X_1, ..., X_n|\theta)\pi(\theta)d\theta = \int \mathcal{L}_n\left(\theta\right)\pi\left(\theta\right)d\theta$$

is the normalizing constant, which is also called the evidedence (Liu and Wasserman, 2014).

3.2.5.2 Bernoulli Distribution

Bernoulli distribution pada penelitian digunakan untuk ..., dibuat dengan library scipy dengan pada modul scipy.stats.bernoulli

The Bernoulli distribution is a discrete distribution having two possible outcomes labelled by $n = 0$ and $n = 1$ in which $n = 1$ ("success") occurs with probability p and $n = 0$ ("failure") occurs with probability $q = 1 - p$, where $0 < p < 1$. It therefore has probability density function

$$p(n) = p^n(1-p)^{1-n}.$$

The distribution of heads and tails in coin tossing is an example of a Bernoulli distribution with $p = q = 1/2$. The Bernoulli distribution is the simplest discrete distribution, and it the building block for other more complicated discrete distributions (Weisstein, 2018).

3.2.5.3 Pymc3

Pymc3 in analyzing test results of this research is used as probabilistic programming tools for bayesian inference method to build probabilistic models. Probabilistic programming offers an effective way to build complex models and allows us to focus more on model design, evaluation, and interpretation, and less on mathematical or computational details (Martin, 2016).

3.2.5.4 Metropolis-Hastings Algorithm

Metropolis-Hastings algorithm (MH) is one of special forms of the Hastings algorithm. The Hastings algorithm (HA) is a stochastic sampling technique widely used throughout computational science. As a Markov Chain Monte Carlo method, HA does not attempt to generate a sequence of independent samples from a "target distribution" $\pi(\cdot)$, defined on the state space(E, ε), but rather a Markov chain $\{Xn, n = 1, 2, 3, ...\}$ having $\pi(\cdot)$ as its invariant distribution. Although variates in the chain are not independent, they may nonetheless be used to estimate statistical expectations with respect to $\pi(\cdot)$.

Let U(0, 1) represent the uniform distribution on (0, 1). In order to use all subsequently described algorithms, given $X_n = x$, we require a "proposal density" $\gamma(\cdot|x)$ which may (or may not) depend on x, and whose variates can be generated by other means. Given $X_n = x \sim \pi(\cdot)$, we can generate $X_{n+1} \sim \pi(\cdot)$ by

Algoritma 3.2 Metropolis-Hastings algorithm

1. Generate $y \sim \gamma(\cdot|x)$ and $r \sim U$(0, 1)
2. If $r \leq \alpha_{HA}(x, y)$, output $X_{n+1} = y$
3. Else, output $X_{n+1} = x$

where densities are assumed to be symmetric (that is, $\gamma(x|y) = \gamma(y|x)$) and the acceptance probability in step 2 is $\alpha_{HA}(x, y) = min\left\{\frac{p(y)}{p(x)}, 1\right\}$ (Minh and Minh, 2015).

Chapter 4

RESULTS AND DISCUSSION

4.1 Identifying Problems

PT Mekar atau Sampoerna Wirausaha adalah perusahan fintech yang menjalankan bisnis dengan platform pinjaman peer-to-peer. Dalam pinjam meminjam terdapat peminjam dan pemberi dana. Mekar Go adalah website yang digunakan untuk menghimpun data peminjam. Data yang telah dihimpun pada versi web Mekar GO saat ini menunjukkan konversi peminjam yang melengkapi pengisian data dari awal sampai akhir kurang dari 10%. Presentase konversi ini menurut tim marketing terlalu kecil dan perlu ditingkatkan.

4.2 Defining Website Measurement

4.2.1 Business Objective

Business objective yang diinginkan adalah menambah jumlah peminjam yang akan diberikan pinjaman peer-to-peer lending.

4.2.2 Website Goal

Website goal yang diinginkan diantaranya adalah: meningkatkan jumlah peminjam yang mendaftar, meningkatkan presentase penyelesaian pengisian form pendaftaran oleh peminjam.

4.2.3 Key Performance Metric

Metric yang digunakan sebagai key performance metric adalah: jumlah pendaftar per bulan dan presentase penyelesaian pengisian form oleh peminjam saat pengisian form pendaftaran.

4.2.4 Target Metric

Target yang diharapkan adalah 500 unique visitor yang mendaftar untuk setiap versi dan presentase penyelesaian data di atas 10% untuk setiap versi. Kedua target tersebut diuji dalam satu bulan.

4.3 Developing a Hypothesis

Presentase penyelesaian pengisian form pendaftaran dipengaruhi oleh urutan pengisian form. Apabila diubah akan meningkatkan presentasenya. Solusi yang diajukan adalah melakukan pengujian A/B testing terhadap dua versi baru disertai dengan penyederhanaan dan penyingkatan pengisian data. Asumsi awal, salah satu dari dua versi baru ini akan lebih baik daripada versi yang sebelumnya.

4.4 Developing and Testing Page Variants

4.4.1 Developing Page Variants

Field yang dilakukan perubahan data dari versi yang sebelumnya terdapat pada tabel 4.1. Bagian perubahan data dikategorikan menjadi dua yaitu personal detail information dan jaminan. Jenis perubahan yang dilakukan adalah removing data untuk data yang dianggap tidak dibutuhkan. Optional adalah bagian yang diubah menjadi tidak wajib diisi, tapi fieldnya masih ada. Yang terakhir adalah new untuk field baru yang tidak ada pada versi sebelumnya. Perubahan field ini berlaku untuk seluruh versi A, B, dan C.

Perubahan yang dilakukan pada personal detail information adalah meremove field alamat usaha atau tempat bekerja, bidang jenis usaha atau jenis pekerjaan, lama usaha atau lama bekerja, jumlah karyawan, foto tempat usaha, status karyawan. Perubahan field yang dijadikan field optional pada bagian personal detail information adalah field email dan foto KTP. Field baru yang ditambahkan pada bagian personal detail information adalah field tanggal lahir. Lalu, bagian jaminan hanya dilakukan perubahan field menjadi optional pada field foto tanah dan bangunan serta foto kendaraan bermotor.

Table 4.1: Data fields changes

Part	Fields Name	Remove	Optional	New
Personal Detail Information	Alamat Usaha / Tempat Bekerja (Provinsi, Kota, Kecamatan, Kelurahan)	v		
	Bidang Jenis Usaha / Jenis Pekerjaan	v		
	Lama Usaha / Lama Bekerja	v		
	Jumlah Karyawan	v		
	Foto Tempat Usaha	v		
	Status Karyawan	v		
	Email		v	
	Foto KTP		v	
	Tanggal Lahir			v
Jaminan	Foto Tanah + Bangunan		v	
	Foto Kendaraan Bermotor		v	

Alur pengisian data untuk versi A, B, dan C didefinisikan pada tabel 4.2. Setiap halaman dianggap juga sebagai langkah atau step dan diberi nama untuk setiap stepnya. Step pengisian data seluruh versi terdiri dari: survey needs page, detail needs page, personal information page, detail loan access page, ekstra pesonal detail information page, dan congrats page. Versi B dan C tidak terdapat step ekstra pesonal detail information page karena disatukan pada step pesonal detail information page. Url untuk seluruh step hampir sama, yang menjadi pembeda adalah suffik alfabet a atau b atau c setelah /ukm/ pada definisi urlnya. Pembedaan suffix ini untuk memudahkan saat membedakan traffic di Google Analytics dashboard.

Table 4.2: Url definition

Step	Site A (/ukm/a/)	Site B (/ukm/b/)	Site C (/ukm/c/)
Survey Needs Page	/ukm/a/survei	/ukm/b/survei	/ukm/c/survei
Detail Needs Page	/ukm/a/detil-survei	/ukm/b/detil-survei	/ukm/c/detil-survei
Personal Information Page	/ukm/a/data-diri	/ukm/b/data-diri	/ukm/c/data-diri
Thanks Page	/ukm/a/terima-kasih	/ukm/b/terima-kasih	/ukm/c/terima-kasih
Detail Loan Access Page	/ukm/a/ pinjaman	/ukm/b/ pinjaman	/ukm/c/ pinjaman
Ekstra Pesonal Detail Information Page	/ukm/a/ tambahan-data-diri	-	-
Congrats Page	/ukm/a/selamat	/ukm/b/selamat	/ukm/c/selamat

Pembeda dari A/B testing ini adalah alur pengisian data dari seluruh versi. Versi A urutan pengisian datanya sama dengan versi sebelumnya. Sedangkan, versi B dan C adalah versi baru yang akan dibandingkan hasilnya manakah yang lebih baik. Adapun urutan alur pengisian data untuk setiap versi adalah sebagai berikut:

- Alur pengisian data versi A: survey needs page - detail needs page - personal information page - thanks page - detail loan access page - ekstra pesonal detail information page - congrats page.
- Alur pengisian data versi B: personal information page - detail loan access page - thanks page - survey needs page - detail needs page - congrats page.
- Alur pengisian data versi C: detail loan access page - personal information page - thanks page - survey needs page - detail needs page -

congrats page.

Pembagian traffic agar seimbang untuk versi A, B, dan C dilakukan di bagian server. Bahasa pemrograman bagian server yang digunakan adalah bahasa pemrograman python. Syntax implementasi pembagian traffic terdapat pada listing 4.1. Logika yang diimplementasikan adalah: pengunjung pertama akan diarahkan ke versi A. Lalu, untuk pengunjung berikutnya, dilakukan query ke database untuk mengetahui versi apa yang terakhir di kunjungi. Jika versi A adalah yang terakhir yang dikunjungi, maka pengunjung selanjutnya akan diarahkan ke versi B. Jika versi B adalah yang terakhir yang dikunjungi, maka pengunjung selanjutnya akan diarahkan ke versi C. Jika versi C adalah yang terakhir yang dikunjungi, maka pengunjung selanjutnya akan diarahkan ke versi A.

Listing 4.1: Traffic splitting

```python
class RoundRobin(models.Model):
    flow_type = models.CharField(max_length=1, null=True)
    partner_slug = models.SlugField(null=True)
    def next(self):
        latest = self.flow_type
        if latest == 'a':
            return 'b'
        elif latest == 'b':
            return 'c'
        elif latest == 'c':
            return 'a'

if RoundRobin.objects.count() == 0:
    next_flow = 'a'
else:
    latest = RoundRobin.objects.last()
    next_flow = latest.next()
```

4.4.2 Testing Page Variants

Testing page variant dilakukan dengan bantuan Google Analytics. Script Google Analytics diterapkan pada setiap step halaman untuk seluruh versi. Syntax script Google Analytic yang digunakan terdapat pada listing program

4.2. Pada syntax “gtag(’config’, ’UA-70057602-1’);” merupakan penanda tracking yang dimiliki Mekar GO. Code unik UA-70057602-1 didapat dari dashboard Google Analytic pada akun Google PT Mekar.

Listing 4.2: Google Analytics implementation

```
<!-- Global Site Tag (gtag.js) - Google Analytics -->
<script async src="https://www.googletagmanager.com/gtag/
    js?id=UA-70057602-1">
</script>
<script>
  window.dataLayer = window.dataLayer || [];
  function gtag(){dataLayer.push(arguments);}
  gtag('js', new Date());
  gtag('config', 'UA-70057602-1');
</script>
```

Tracking Google Analytic dilakukan dari tanggal 24 September 2017 sampai 23 Oktober 2017. Traffic tinggi terjadi pada tanggal 3 September 2017, 9-10 Oktober 2017, 13 Oktober 2017 dan 16 Oktober 2017. Grafik fluktuasi traffic pengunjung website Mekar GO pada figure 4.1 didapat dari dashboard akun Google Analytics PT Mekar. Traffic yang ditunjukkan adalah traffic gabungan untuk seluruh versi A, B, dan C.

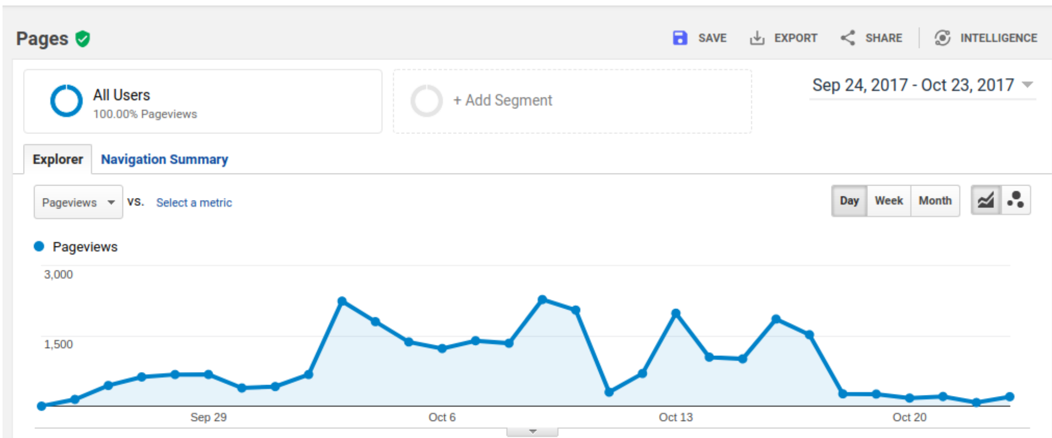


Figure 4.1: Graphic google analytics mekargo.id

Di bawah grafik traffic terdapat tabel detail statistik pengunjung pada setiap halaman. Tabel statistik pengunjung yang terdapat pada dashboard Google Analytic terlihat seperti pada figure 4.2. Tidak seluruh data yang terdapat pada tabel digunakan untuk proses analysis A/B testing. Row yang digunakan dari tabel statistik pengunujung adalah row unique pageviews. Hal ini agar sesuai dengan target metric yang mengukur unique visitor. Figure 4.2 menunjukkan bahwa tiga data awal unique visitor pada versi A, B dan C sudah memnuhi target unique visitor lebih besar dari 500.

Primary Dimension: Page | Page Title | Other

Plot Rows | Secondary dimension | Sort Type: Default | /ukm/ | advanced

Page	Pageviews	Unique Pageviews	Avg. Time on Page	Entrances	Bounce Rate	% Exit	Page Value
	27,335 % of Total: 97.94% (27,911)	19,711 % of Total: 97.72% (20,170)	00:01:23 Avg for View: 00:01:23 (0.13%)	10,343 % of Total: 98.15% (10,538)	54.08% Avg for View: 53.77% (0.59%)	37.85% Avg for View: 37.76% (0.25%)	$0.00 % of Total: 0.00% ($0.00)
1. /ukm/c/	5,164 (18.89%)	3,801 (19.28%)	00:00:55	3,709 (35.86%)	56.70%	56.66%	$0.00 (0.00%)
2. /ukm/b/	4,737 (17.33%)	3,385 (17.17%)	00:01:02	3,281 (31.72%)	52.45%	51.61%	$0.00 (0.00%)
3. /ukm/a/	4,222 (15.45%)	3,123 (15.84%)	00:00:52	3,039 (29.38%)	52.45%	50.76%	$0.00 (0.00%)
4. /ukm/b/data-diri/	2,200 (8.05%)	1,497 (7.59%)	00:03:30	76 (0.73%)	51.32%	30.32%	$0.00 (0.00%)
5. /ukm/c/pinjaman/	1,894 (6.93%)	1,494 (7.58%)	00:01:35	35 (0.34%)	71.43%	38.01%	$0.00 (0.00%)
6. /ukm/a/survei/	2,270 (8.30%)	1,392 (7.06%)	00:00:37	44 (0.43%)	54.55%	10.66%	$0.00 (0.00%)
7. /ukm/a/detil-survei/	1,913 (7.00%)	1,156 (5.86%)	00:00:31	24 (0.23%)	50.00%	6.33%	$0.00 (0.00%)
8. /ukm/a/data-diri/	1,725 (6.31%)	1,127 (5.72%)	00:02:30	22 (0.21%)	59.09%	14.38%	$0.00 (0.00%)
9. /ukm/a/pinjaman/	838 (3.07%)	751 (3.81%)	00:02:14	22 (0.21%)	68.18%	30.91%	$0.00 (0.00%)
10. /ukm/b/pinjaman/	569 (2.08%)	501 (2.54%)	00:02:21	26 (0.25%)	42.31%	36.20%	$0.00 (0.00%)

Show rows: 10 Go to: 1 1 - 10 of 21

This report was generated on 1/31/18 at 7:05:00 PM - Refresh Report

Figure 4.2: Table google analytics mekargo.id

4.5 Analyzing Test Results

Data yang didapat dari google analytics dilakukan pemrosesan dan analisa dengan metode bayesian inference dengan pymc3.

```
In [1]: import pymc3 as pm
        import numpy as np
        %matplotlib inline
        from IPython.core.pylabtools import figsize
        import matplotlib.pyplot as plt
        import scipy.stats as stats
        figsize(12.5, 4)
```

Figure 4.3: Import python library

Syntax pertama untuk menganalisa hasil A/B tesing pada penelitian ini adalah mengimport seluruh library python yang dibutuhkan yaitu pymc3, numpy, Ipython, matplotlib, dan scipy. Syntax ini ditulis pada environment jupyter notebook. Syntaxnya terdapat pada figure 4.3.

```
In [2]: true_A = 120
        true_B = 129
        true_C = 84

        N_sample = 1156
```

Figure 4.4: Initializing sample

Figure 4.4 berisi syntax inisialisasi nilai awal true_A, true_B, true_C, dan N_sample. Nilai awal true_A, true_B, dan true_C berfungsi sebagai variabel penyimpan nilai jumlah user yang menyelesaikan proses penyelesaian data pada versi A, B, dan C. Sedangkan, N_sample sebagai variabel penyimpan banyaknya sample yang digunakan. Nilai awal true_A sebanyak 120 user yang menyelesaikan proses pengisian data pada versi A, nilai awal true_B sebanyak 129 user yang menyelesaikan proses pengisian data pada versi B, dan nilai awal true_C sebanyak 84 user yang menyelesaikan proses pengisian data pada versi C. Sedangkan, N_sample sebanyak 1156 user sebagai sampel yang melakukan proses pengisian data pada setiap versi A, B, dan C. Sampel yang digunakan nilainya sama untuk seluruh versi.

```
In [3]: true_p_A = true_A/float(N_sample)
        true_p_B = true_B/float(N_sample)
        true_p_C = true_C/float(N_sample)

        print("true p_A:", true_p_A)
        print("true p_B:", true_p_B)
        print("true p_C:", true_p_C)

        true p_A: 0.10380622837370242
        true p_B: 0.1115916955017301
        true p_C: 0.0726643598615917
```

Figure 4.5: Calculating probability

Syntax operasi perhitungan yang terdapat pada figure 4.5 merupakan langkah pertama pada algoritma bayesian inference 3.1. Hasil operasi syntax ini menentukan nilai probability density atau yang disebut juga sebagai prior distribution yang menggambarkan beliefs about a parameter. Dalam hal ini mengenai jumlah user yang menyelesaikan pengisian data dibandingkan jumlah total user yang melakukan pengisian data (yang menyelesaikan ditambah yang tidak). Variable true_p_A merupakan prior distribution untuk versi A, variable true_p_B merupakan prior distribution untuk versi B, dan variable true_p_C merupakan prior distribution untuk versi C. Adapun nilai prior distribution untuk masing-masing versi adalah sebagai berikut:

true_p_A = 0.10380622837370242
true_p_B = 0.1115916955017301
true_p_C = 0.0726643598615917

Nilai true_p_A, true_p_B, dan true_p_C didapat dengan melakukan operasi perhitungan terhadap nilai inisiasi awal yang didapat sebelumnya. Nilai ini menunjukan presentase penyelesaian pengisian data sampai akhir yang berhasil untuk setiap versi. Variabel true_p_A menunjukan presentase penyelesaian pengisian data versi A sebanyak 10,38%. Variabel true_p_B menunjukan presentase penyelesaian pengisian data versi A sebanyak 11,16%. Variabel true_p_C menunjukan presentase penyelesaian pengisian data versi A sebanyak 7,27%. Versi A dan B sudah memenuhi target metrik yang menginginkan presentase penyelesaian pengisian data di atas 10 persen.

```
In [4]: observations_A = stats.bernoulli.rvs(true_p_A, size=N_sample)
        observations_B = stats.bernoulli.rvs(true_p_B, size=N_sample)
        observations_C = stats.bernoulli.rvs(true_p_C, size=N_sample)

        print("Obs from Site A: ", observations_A[:30], "...")
        print("Obs from Site B: ", observations_B[:30], "...")
        print("Obs from Site C: ", observations_C[:30], "...")

        Obs from Site A:  [0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0] ...
        Obs from Site B:  [0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0] ...
        Obs from Site C:  [0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0] ...
```

Figure 4.6: Bernoulli observation

Tahap ini merupakan langkah kedua pada algoritma bayesian inference 3.1 yaitu memilih statistical model yang reflects beliefs dengan prior distribution yang dipilih pada langkah sebelumnya. Penelitian ini memilih statistical model bernoulli distribution untuk merefleksikan keyakinan terhadap prior distribution pada langkah sebelumnya. Implementasi pada syntax pythonnya menggunakan method stats yang diimport dari library scipy. Hasil dari bernoulli distributionnya adalah numpy array yang berisi angka random 0 dan 1 yang digenerate sesuai prior distribution untuk masing-masing versi A, B, dan C. Angka 1 mewakili nilai true dan angka 0 mewakili nilai false. Nilai bernoulli distribution yang dihasilkan dari scipy dipengaruhi oleh inputan true_p_A, true_p_B, true_p_C dan N_sample. Syntax pembuatan statistical model bernoulli distribution dan hasilnya bisa dilihat pada figure 4.6. Nilai bernoulli distribution untuk setiap versi disimpan pada variable observation_A, observation_B, dan observation_C.

```
In [5]: print(np.mean(observations_A))
        print(np.mean(observations_B))
        print(np.mean(observations_C))

        print(np.sum(observations_A))
        print(np.sum(observations_B))
        print(np.sum(observations_C))

0.0951557093426
0.117647058824
0.0726643598616
110
136
84
```

Figure 4.7: Sum and mean of bernoulli observation

Nilai rata-rata dari hasil observasi bernoulli didapat dengan bantuan library numpy dengan method mean. Nilai yang dihasilkan adalah:

mean_p_A = 0.098615916955

mean_p_B = 0.11937716263

mean_p_C = 0.0743944636678

Nilai rata-rata observasi bernoulli ini menjadi pembaruan nilai belief pada langkah ketiga algoritma bayessian inference 3.1. Nilai rata-rata observasi bernoulli disebut juga nilai true frequency. Nilai true frequency pada kasus ini adalah nilai peluang sebenarnya user menyelesaikan pengisian data sampai akhir yang bernilai true. Nilai sum observasi bernoulli menunjukkan berapa banyak nilai true yang dihasilkan pada distribusi. Adapun banyaknya nilai true dari setiap versi adalah sebagai berikut: versi A diwakili variabel sum_true_p_A sebanyak 114 kali true, versi B diwakili variabel sum_true_p_B sebanyak 138 kali true, dan versi C diwakili variabel sum_true_p_C sebanyak 86 kali true. Setelah itu, melakukan perhitungan posterior distribution.

```
In [6]: with pm.Model() as model:
            p_A = pm.Uniform("p_A", 0, 1)
            p_B = pm.Uniform("p_B", 0, 1)
            p_C = pm.Uniform("p_C", 0, 1)

            # Define the deterministic delta function. This is our unknown of interest.
            delta_A_B = pm.Deterministic("delta_A_B", p_A - p_B)
            delta_A_C = pm.Deterministic("delta_A_C", p_A - p_C)
            delta_B_C = pm.Deterministic("delta_B_C", p_B - p_C)

            # Set of observations, in this case we have three observation datasets.
            obs_A = pm.Bernoulli("obs_A", p_A, observed=observations_A)
            obs_B = pm.Bernoulli("obs_B", p_B, observed=observations_B)
            obs_C = pm.Bernoulli("obs_C", p_C, observed=observations_C)

            step = pm.Metropolis()
            trace = pm.sample(20000, step=step)
            burned_trace=trace[1000:]

100%|██████████| 20500/20500 [00:31<00:00, 643.38it/s]
```

Figure 4.8: Pymc probabilistic model

Perhitungan posterior distribution dengan algorithma MH 3.2 pada pymc probability models. Syntaxnya terdapat pada figure 4.8. Syntax pada bagian ini tidak mencerminkan secara runtut sesuai algoritma Metropolis-Hastings 3.2. Syntax yang ditampilkan hanya memberikan nilai inputan apa saja yang diperlukan disertai dengan penggunaan metode probability models dari pymc yang digunakan. Proses algoritma dilakukan di balik layar oleh metode probability models dari pymc. Jadi, algoritma Metropolis-Hastings dijalankan di balik layar oleh metode pm.Metropolis(). Sample algoritma Metropolis-Hastings yang digunakan sebanyak 2000. Arti dari trace[1000:] artinya diambil nilai sampling dari nilai ke 1000 sampai terakhir, hal ini untuk mengambil 95% sample dan mengabaikan 0,05 (1000 dibagi 20000) sample awal karena dianggap belum konvergen.

```
In [7]: p_A_samples = burned_trace["p_A"]
        p_B_samples = burned_trace["p_B"]
        p_C_samples = burned_trace["p_C"]
        delta_A_B_samples = burned_trace["delta_A_B"]
        delta_A_C_samples = burned_trace["delta_A_C"]
        delta_B_C_samples = burned_trace["delta_B_C"]
```

Figure 4.9: Pymc probabilistic model results

Output dari figure 4.8 adalah hasil posterior distribution yang dibutuhkan untuk proses analisa. Output tersebut diekstrak ke dalam beberapa variable, seperti pada figure 4.9. Variabel p_A_samples adalah nilai akhir posterior distribution versi A. Variabel p_B_samples adalah nilai akhir posterior distribution versi B. Variabel p_C_samples adalah nilai akhir posterior distribution versi C. Variabel delta_A_B_samples adalah nilai akhir posterior distribution perbandingan antara versi A dan versi B. Variabel delta_A_C_samples adalah nilai akhir posterior distribution perbandingan antara versi A dan versi C. Variabel delta_B_C_samples adalah nilai akhir posterior distribution perbandingan antara versi B dan versi C.

```
In [8]: figsize(12.5, 10)

        #histogram of posteriors

        ax = plt.subplot(311)

        plt.xlim(.06, .2)
        plt.hist(p_A_samples, histtype='stepfilled', bins=25, alpha=0.85,
                 label="posterior of $p_A$", color="#A60628", normed=True)
        plt.vlines(true_p_A, 0, 80, linestyle="--", label="true $p_A$ (unknown)")
        plt.legend(loc="upper right")
        plt.title("Posterior distributions of $p_A$, $p_B$, and delta unknowns")

Out[8]: Text(0.5,1,'Posterior distributions of $p A$, $p B$, and delta unknowns')
```

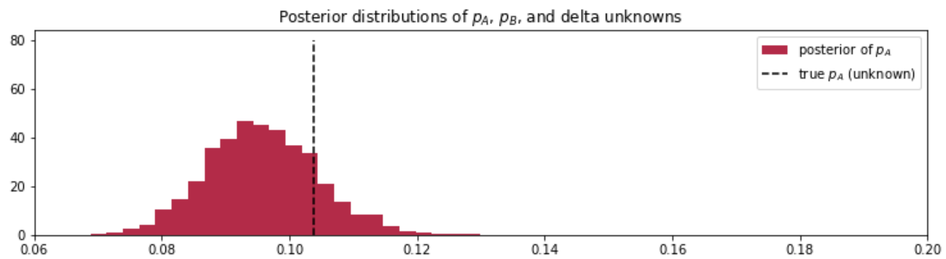


Figure 4.10: Posteriror distribution P_A

Figure 4.10 merupakan visualisasi posterior distribution probabilistik untuk versi A. Axis x diplot pada range 0,06 sampai 0,2, Akan tetapi distribusi nilai posterior untuk versi A berada diantara 0,07 sampai 0,13. Garis vertikal putus-putus merupakan letak nilai prior distribution versi A pada variabel true_p_A = 0.10380622837370242 yang sudah dideklarasikan pada bagian awal. Nilai posterior terbanyak terdapat pada sekitar 0,092.

```
In [9]: ax = plt.subplot(312)

        plt.xlim(.06, .2)
        plt.hist(p_B_samples, histtype='stepfilled', bins=25, alpha=0.85,
                 label="posterior of $p_B$", color="#467821", normed=True)
        plt.vlines(true_p_B, 0, 80, linestyle="--", label="true $p_B$ (unknown)")
        plt.legend(loc="upper right")

Out[9]: <matplotlib.legend.Legend at 0x7f03334969e8>
```

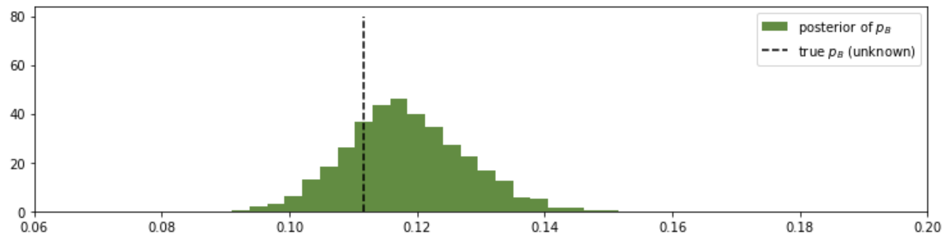


Figure 4.11: Posteriror distribution P_B

Figure 4.11 merupakan visualisasi posterior distribution probabilistik untuk versi B. Axis x diplot pada range 0,06 sampai 0,2. Akan tetapi, distribusi nilai posterior untuk versi B berada diantara 0,09 sampai 0,15. Garis vertikal putus-putus merupakan letak nilai prior distribution versi B pada variabel true_p_B = 0.1115916955017301 yang sudah dideklarasikan pada bagian awal. Nilai posterior terbanyak terdapat pada sekitar 0,118.

```
In [10]: ax = plt.subplot(313)

         plt.xlim(0, .1)
         plt.hist(p_C_samples, histtype='stepfilled', bins=25, alpha=0.85,
                  label="posterior of $p_C$", color="#D6F841", normed=True)
         plt.vlines(true_p_C, 0, 80, linestyle="--", label="true $p_C$ (unknown)")
         plt.legend(loc="upper right")

Out[10]: <matplotlib.legend.Legend at 0x7f0332445a90>
```

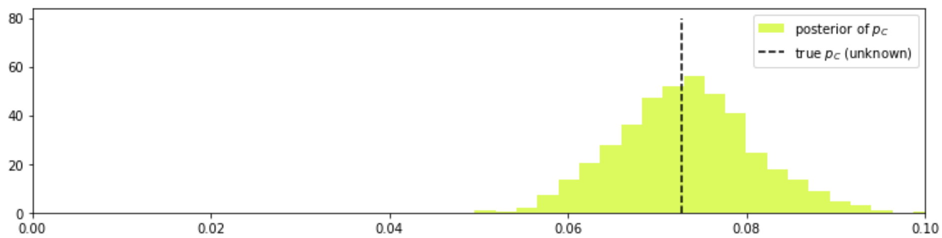


Figure 4.12: Posteriror distribution P_C

Figure 4.12 merupakan visualisasi posterior distribution probabilistik untuk versi C. Axis x diplot pada range 0,00 sampai 0,1. Akan tetapi, distribusi nilai posterior untuk versi C berada diantara 0,05 sampai 0,09. Garis vertikal putus-putus merupakan letak nilai prior distribution versi C pada variabel true_p_C = 0.0726643598615917 yang sudah dideklarasikan pada bagian awal. Nilai posterior terbanyak terdapat pada sekitar 0,072.

Beberapa grafik selanjutnya (figure 4.13, 4.14, dan 4.15) merupakan visualisasi posterior distribution kombinasi perbandingan seluruh versi. Kombinasi perbandingan dilakukan untuk setiap dua versi. Kombinasi perbandingan yang dihasilkan adalah perbandingan antara versi A dan B disebut delta A-B, perbandingan antara versi A dan C disebut juga delta A-C, dan perbandingan antara versi B dan C disebut juga delta B-C.

In [11]:
```
ax = plt.subplot(313)
plt.hist(delta_A_B_samples, histtype='stepfilled', bins=30, alpha=0.85,
         label="posterior of delta A-B", color="#7A68A6", normed=True)
plt.vlines(true_p_A - true_p_B, 0, 60, linestyle="--",
           label="true delta (unknown)")
plt.vlines(0, 0, 60, color="black", alpha=0.2)
plt.legend(loc="upper right");
```

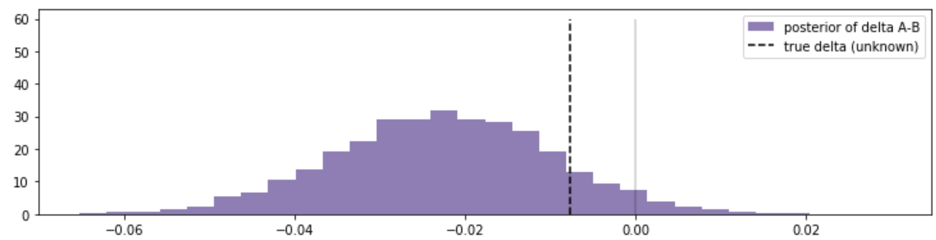


Figure 4.13: Posteriror distribution delta A-B

Figure 4.13 merupakan visualisasi posterior distribution probabilistik untuk delta A-B. Distribusi nilai posterior untuk delta A-B berada diantara -0,05 sampai 0,02. Garis vertikal putus-putus merupakan letak nilai prior distribution delta A-B berada pada nilai sekitar -0.01. Garis vertikal tanpa putus-putus menunjukkan nilai 0 pada axis x. Persebaran posterior distribution lebih banyak berada pada sebelah kiri nilai 0 atau pada nilai negatif. Artinya, variabel pengurang dari perbandingan delta A-B menjadi versi yang lebih baik. Versi B menjadi versi yang lebih baik dibandingkan versi A karena versi B sebagai pengurang pada delta A-B.

In [12]:
```
ax = plt.subplot(313)
plt.hist(delta_A_C_samples, histtype='stepfilled', bins=30, alpha=0.85,
         label="posterior of delta A-C", color="#B868B6", normed=True)
plt.vlines(true_p_A - true_p_C, 0, 60, linestyle="--",
           label="true delta (unknown)")
plt.vlines(0, 0, 60, color="black")
plt.legend(loc="upper right");
```

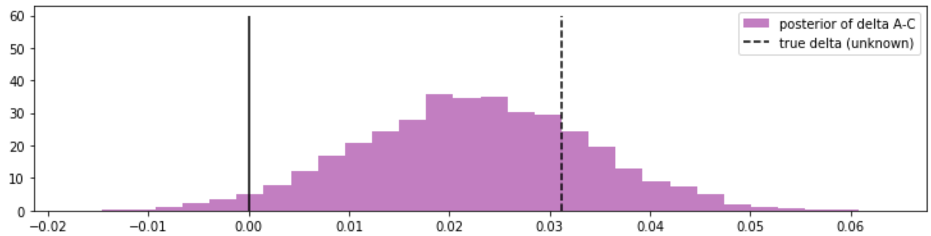


Figure 4.14: Posteriror distribution delta A-C

Figure 4.14 merupakan visualisasi posterior distribution probabilistik untuk delta A-B. Distribusi nilai posterior untuk delta A-C berada diantara -0,015 sampai 0,06. Garis vertikal putus-putus merupakan letak nilai prior distribution delta A-C berada pada nilai sekitar 0.031. Garis vertikal tanpa putus-putus menunjukkan nilai 0 pada axis x. Persebaran posterior distribution lebih banyak berada pada sebelah kanan nilai 0 atau pada nilai positif. Artinya, variabel yang dikurangi dari perbandingan delta A-C menjadi versi yang lebih baik. Versi A menjadi versi yang lebih baik dibandingkan versi C karena versi A yang dikurangi pada delta A-C.

In [13]:
```
ax = plt.subplot(313)
plt.hist(delta_B_C_samples, histtype='stepfilled', bins=30, alpha=0.85,
         label="posterior of delta B-C", color="#24B8A6", normed=True)
plt.vlines(true_p_B - true_p_C, 0, 60, linestyle="--",
           label="true delta (unknown)")
plt.vlines(0, 0, 60, color="black")
plt.legend(loc="upper right");
```

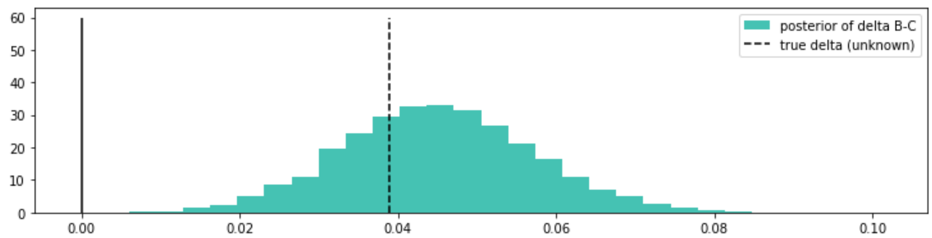


Figure 4.15: Posteriror distribution delta B-C

Figure 4.13 merupakan visualisasi posterior distribution probabilistik untuk delta A-B. Distribusi nilai posterior untuk delta A-B berada diantara 0,01 sampai 0,082. Garis vertikal putus-putus merupakan letak nilai prior distribution delta A-B berada pada nilai sekitar 0.038. Garis vertikal tanpa putus-putus menunjukkan nilai 0 pada axis x. Persebaran posterior distribution seluruhnya berada pada sebelah kanan nilai 0 atau pada nilai positif. Artinya, variabel yang dikurangi dari perbandingan delta B-C menjadi versi yang lebih baik. Versi B menjadi versi yang lebih baik dibandingkan versi C karena versi B yang dikurangi pada pada delta B-C.

```
In [14]: # Count the number of samples less than 0, i.e. the area under the curve
         # before 0, represent the probability that site A is worse than site B.
         print("Probability site A is WORSE than site B: %.3f" % \
             np.mean(delta_A_B_samples < 0))

         print("Probability site A is BETTER than site B: %.3f" % \
             np.mean(delta_A_B_samples > 0))

         print("\nProbability site A is WORSE than site C: %.3f" % \
             np.mean(delta_A_C_samples < 0))

         print("Probability site A is BETTER than site C: %.3f \n" % \
             np.mean(delta_A_C_samples > 0))

         print("Probability site B is WORSE than site C: %.3f" % \
             np.mean(delta_B_C_samples < 0))

         print("Probability site B is BETTER than site C: %.3f" % \
             np.mean(delta_B_C_samples > 0))

         Probability site A is WORSE than site B: 0.961
         Probability site A is BETTER than site B: 0.039

         Probability site A is WORSE than site C: 0.025
         Probability site A is BETTER than site C: 0.975

         Probability site B is WORSE than site C: 0.000
         Probability site B is BETTER than site C: 1.000
```

Figure 4.16: Final result combination of all variants

Hasil analisa perbandingan postireior distribution sebelumnya sudah mengungkapkan bagaimana peringkat perbandingan seluruh versi. Figure 4.16 menunjukan nilai presentase keyakinan untuk seluruh kombinasi perbandingan. Versi A dibanding versi B pemenangnya adalah versi B, dengan keyakinan versi A lebih buruk 96,1 persen (0,961) persen dari versi B. Versi A dibanding versi C pemenangnya adalah versi A, dengan keyakinan versi A lebih baik 97,5 persen (0,975) dari versi C. Versi B dibanding versi C pemenangnya adalah versi B, dengan keyakinan versi B lebih baik 100 persen (1,00) dari versi C. Jadi, versi yang paling baik adalah versi B karena versi B $>$ versi A $>$ versi C.

Chapter 5

CONCLUSION AND SUGGESTIONS

5.1 Conclusion

Metode penelitian yang dilakukan untuk melakukan A/B testing pada website mekargo.id meliputi: identifying problems, defining website measurement, developing a hypothesis, developing and testing page variants, dan analyzing test results. Cara menganalis hasil A/B testing dengan metode bayesian inference dan tools pymc3 dilakukan melalui Metropolis-Hastings algorithm sebagai teknik sampling, ipython dan jupyter notebook sebagai environment pemrograman, scipy untuk memopulasi bernoulli distribution, numpy sebagai penyimpan numerikal data berbentuk array, dan matplotlib sebagai penampil grafik hasil pengolahan data. Dari hasil akhir seluruh kombinasi probabilitas yang dihasilkan dapat disimpulkan bahwa versi B adalah hasil yang terbaik karena selalu lebih baik saat dibandingkan dengan versi A dan C.

5.2 Suggestions

Penelitian berikutnya dapat dicoba untuk melakukan A/B testing dengan versi lebih banyak dan pengembangan metode bayesian inference dengan algoritma multi-arm bandit.

Bibliography

Al-Rfou, R., Alain, G., Almahairi, A., Angermueller, C., Bahdanau, D., Ballas, N., Bastien, F., Bayer, J., Belikov, A., Belopolsky, A., et al. (2016). Theano: A python framework for fast computation of mathematical expressions. *arXiv preprint*.

CXL (2017). *How to Build a Strong A/B Testing Plan That Gets Results*. [Online]. Available at: https://conversionxl.com/blog/how-to-build-a-strong-ab-testing-plan-that-gets-results/ [Accessed 17 February 2018].

Davidson-Pilon, C. (2015). *Bayesian Methods for Hackers: Probabilistic Programming and Bayesian Inference*. Addison-Wesley Data & Analytics Series. Pearson Education.

Dixon, E., Enos, E., and Brodmerkle, S. (2015). A/b testing. US Patent 9,077,780.

Fandango, A. and Idris, I. (2017). *Python Data Analysis - Second Edition*. Packt Publishing.

Fernández-i Marín, X. et al. (2016). ggmcmc: Analysis of mcmc samples and bayesian inference. *Journal of Statistical Software*, 70(9).

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2014). *Bayesian data analysis*, volume 2. CRC press Boca Raton, FL.

Goldberg, D. and Johndrow, J. E. (2017). A decision theoretic approach to a/b testing. *arXiv preprint arXiv:1710.03410*.

Hunter, J., Dale, D., Firing, E., and Droettboom, M. (2014). The matplotlib development team. matplotlib: Python plotting – documentation. 2013.

Jiménez, J. and Ginebra, J. (2017). pygpgo: Bayesian optimization for python. *The Journal of Open Source Software*.

Johari, R., Pekelis, L., and Walsh, D. J. (2015). Always valid inference: Bringing sequential analysis to a/b testing. *arXiv preprint arXiv:1512.04922*.

Jones, E., Oliphant, T., and Peterson, P. (2014). {SciPy}: open source scientific tools for {Python}.

Kaufmann, E., Cappé, O., and Garivier, A. (2014). On the complexity of a/b testing. In *Conference on Learning Theory*, pages 461–481.

Kelsey, T. (2017). *Introduction to Google Analytics: A Guide for Absolute Beginners*. Apress.

Kemenkeu (2015). *Peran Penting UKM Dorong Perekonomian Indonesia*. [Online]. Available at: https://www.kemenkeu.go.id/Berita/peran-penting-ukm-dorong-perekonomian-indonesia [Accessed 17 January 2018].

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B. E., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J. B., Grout, J., Corlay, S., et al. (2016). Jupyter notebooks-a publishing format for reproducible computational workflows. In *ELPUB*, pages 87–90.

Kohavi, R., Deng, A., Frasca, B., Walker, T., Xu, Y., and Pohlmann, N. (2013). Online controlled experiments at large scale. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1168–1176. ACM.

Kohavi, R. and Longbotham, R. (2017). Online controlled experiments and a/b testing. In *Encyclopedia of Machine Learning and Data Mining*, pages 922–929. Springer.

Liu, H. and Wasserman, L. (2014). *Statistical Machine Learning Chapter 12 Bayesian Inference*. [Online]. Available at: http://www.stat.cmu.edu/ larry/=sml/Bayes.pdf [Accessed 17 January 2018].

Lubanovic, B. (2014). *Introducing Python: Modern Computing in Simple Packages*. O'Reilly Media.

Lutz, M. (2013). *Learning Python: Powerful Object-Oriented Programming*. Safari Books Online. O'Reilly Media.

Martin, O. (2016). *Bayesian Analysis with Python*. Packt Publishing.

Mekar (2017). *Mekar FAQ*. [Online]. Available at: https://mekar.id/faq/ [Accessed 17 January 2018].

Meurer, A., Smith, C. P., Paprocki, M., Čertík, O., Kirpichev, S. B., Rocklin, M., Kumar, A., Ivanov, S., Moore, J. K., Singh, S., et al. (2017). Sympy: symbolic computing in python. *PeerJ Computer Science*, 3:e103.

Minh, D. D. and Minh, D. L. (2015). Understanding the hastings algorithm. *Communications in Statistics-Simulation and Computation*, 44(2):332–349.

Omniconvert (2017). *How to create an A/B test*. [Online]. Available at: https://www.omniconvert.com/how-to-create-an-a-b-test [Accessed 17 January 2018].

Pourzanjani, A. A., Jiang, R. M., Atzberger, P. J., and Petzold, L. R. (2017). General bayesian inference over the stiefel manifold via the givens transform. *arXiv preprint arXiv:1710.09443*.

Ragan-Kelley, M., Perez, F., Granger, B., Kluyver, T., Ivanov, P., Frederic, J., and Bussonnier, M. (2014). The jupyter/ipython architecture: a unified view of computational research, from interactive exploration to communication and publication. In *AGU Fall Meeting Abstracts*.

Rougier, N. P., Hinsen, K., Alexandre, F., Arildsen, T., Barba, L. A., Benureau, F. C., Brown, C. T., De Buyl, P., Caglayan, O., Davison, A. P., et al. (2017). Sustainable computational science: the rescience initiative. *PeerJ Computer Science*, 3:e142.

Salvatier, J., Wiecki, T. V., and Fonnesbeck, C. (2016). Probabilistic programming in python using pymc3. *PeerJ Computer Science*, 2:e55.

Smith, A. M., Niemeyer, K. E., Katz, D. S., Barba, L. A., Githinji, G., Gymrek, M., Huff, K. D., Madan, C. R., Mayes, A. C., Moerman, K. M., et al. (2018). Journal of open source software (joss): design and first-year review. *PeerJ Computer Science*, 4:e147.

Strickland, C. M., Burdett, R. L., Denham, R., and Mengersen, K. L. (2014). Pyssm: a python module for bayesian inference of linear gaussian state space models. *Journal of Statistical Software*, 57(6).

Taylor, B., Davies, T., Rowlingson, B., and Diggle, P. (2015). Bayesian inference and data augmentation schemes for spatial, spatiotemporal and multivariate log-gaussian cox processes in r. *Journal of Statistical Software*, 63:1–48.

Van Der Walt, S., Colbert, S., and Varoquaux, G. (2016). The numpy array: a structure for efficient numerical computation (2011).

Vo, T., Czygan, M., et al. (2015). *Getting started with Python data analysis: learn to use powerful Python libraries for effective data processing and analysis*. Packt Publ.

Vohl, D., Barnes, D. G., Fluke, C. J., Poudel, G., Georgiou-Karistianis, N., Hassan, A. H., Benovitski, Y., Wong, T. H., Kaluza, O. L., Nguyen, T. D., et al. (2016). Large-scale comparative visualisation of sets of multidimensional data. *PeerJ Computer Science*, 2:e88.

Weber, J. (2015). *Practical Google Analytics and Google Tag Manager for Developers*. Apress.

Weisstein, E. W. (2018). *Bernoulli Distribution*. [Online]. Available at: http://mathworld.wolfram.com/BernoulliDistribution.html [Accessed 23 February 2018].

Wilson, G., Perez, F., and Norvig, P. (2014). Teaching computing with the ipython notebook. In *Proceedings of the 45th ACM technical symposium on Computer science education*, pages 740–740. ACM.

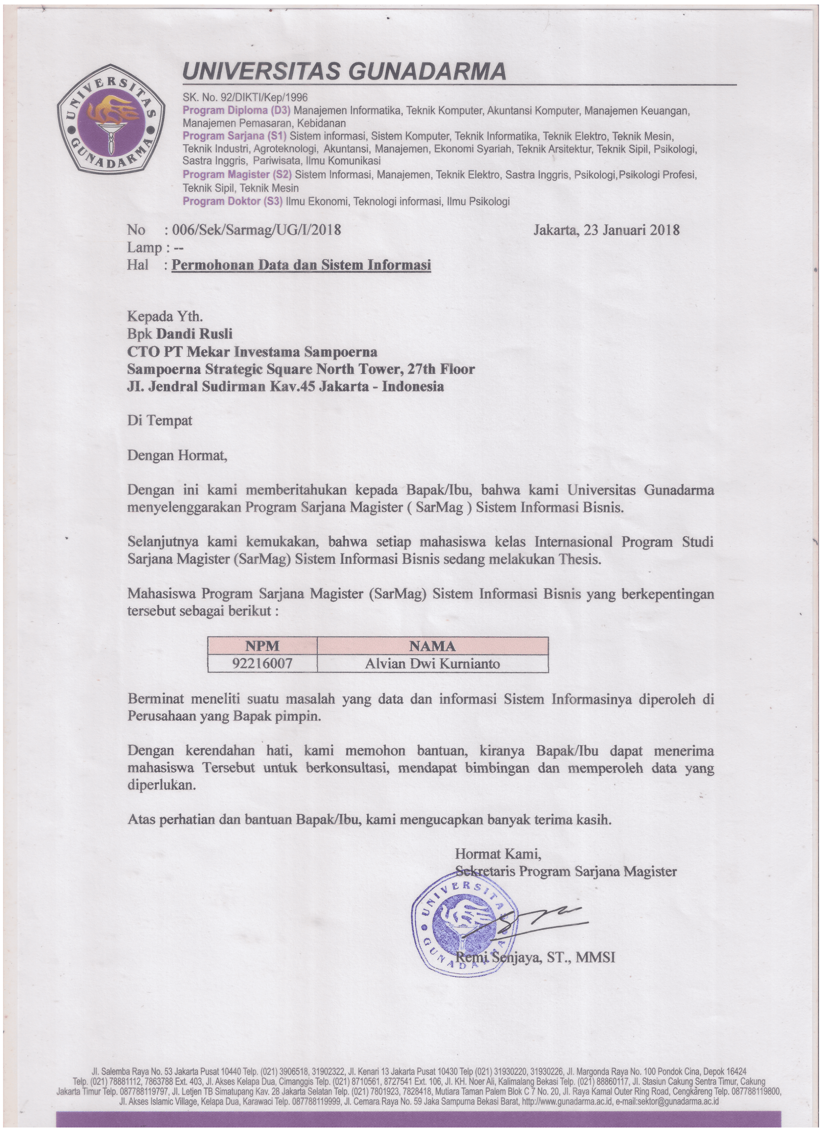
Wilson, J. D. and Uminsky, D. T. (2017). The power of a/b testing under interference. *arXiv preprint arXiv:1710.03855*.

Wuisan, N. Y. (2017). Pengaruh kredit bank terhadap pendapatan pelaku-pelaku umkm (khususnya meubel) di desa leilem, kec. sonder, kab. minahasa. *JURNAL BERKALA ILMIAH EFISIENSI*, 17(01).

Yang, L. and Perrin, J. M. (2014). Tutorials on google analytics: How to craft a web analytics report for a library web site. *Journal of Web Librarianship*, 8(4):404–417.

APPENDIX

Figure A.1: Letter of data and information permission request



UNIVERSITAS GUNADARMA

SK. No. 92/DIKTI/Kep/1996
Program Diploma (D3) Manajemen Informatika, Teknik Komputer, Akuntansi Komputer, Manajemen Keuangan, Manajemen Pemasaran, Kebidanan
Program Sarjana (S1) Sistem informasi, Sistem Komputer, Teknik Informatika, Teknik Elektro, Teknik Mesin, Teknik Industri, Agroteknologi, Akuntansi, Manajemen, Ekonomi Syariah, Teknik Arsitektur, Teknik Sipil, Psikologi, Sastra Inggris, Pariwisata, Ilmu Komunikasi
Program Magister (S2) Sistem Informasi, Manajemen, Teknik Elektro, Sastra Inggris, Psikologi, Psikologi Profesi, Teknik Sipil, Teknik Mesin
Program Doktor (S3) Ilmu Ekonomi, Teknologi informasi, Ilmu Psikologi

No : 006/Sek/Sarmag/UG/I/2018 Jakarta, 23 Januari 2018
Lamp : --
Hal : **Permohonan Data dan Sistem Informasi**

Kepada Yth.
Bpk **Dandi Rusli**
CTO PT Mekar Investama Sampoerna
Sampoerna Strategic Square North Tower, 27th Floor
Jl. Jendral Sudirman Kav.45 Jakarta - Indonesia

Di Tempat

Dengan Hormat,

Dengan ini kami memberitahukan kepada Bapak/Ibu, bahwa kami Universitas Gunadarma menyelenggarakan Program Sarjana Magister (SarMag) Sistem Informasi Bisnis.

Selanjutnya kami kemukakan, bahwa setiap mahasiswa kelas Internasional Program Studi Sarjana Magister (SarMag) Sistem Informasi Bisnis sedang melakukan Thesis.

Mahasiswa Program Sarjana Magister (SarMag) Sistem Informasi Bisnis yang berkepentingan tersebut sebagai berikut :

NPM	NAMA
92216007	Alvian Dwi Kurnianto

Berminat meneliti suatu masalah yang data dan informasi Sistem Informasinya diperoleh di Perusahaan yang Bapak pimpin.

Dengan kerendahan hati, kami memohon bantuan, kiranya Bapak/Ibu dapat menerima mahasiswa Tersebut untuk berkonsultasi, mendapat bimbingan dan memperoleh data yang diperlukan.

Atas perhatian dan bantuan Bapak/Ibu, kami mengucapkan banyak terima kasih.

Hormat Kami,
Sekretaris Program Sarjana Magister

Remi Senjaya, ST., MMSI

Jl. Salemba Raya No. 53 Jakarta Pusat 10440 Telp. (021) 3906518, 31902322, Jl. Kenari 13 Jakarta Pusat 10430 Telp (021) 31930220, 31930226, Jl. Margonda Raya No. 100 Pondok Cina, Depok 16424 Telp. (021) 78881112, 7863788 Ext. 403, Jl. Akses Kelapa Dua, Cimanggis Telp. (021) 8710561, 8727541 Ext. 106, Jl. KH. Noer Ali, Kalimalang Bekasi Telp. (021) 88860117, Jl. Stasiun Cakung Sentra Timur, Cakung Jakarta Timur Telp. 087788119797, Jl. Letjen TB Simatupang Kav. 28 Jakarta Selatan Telp. (021) 7801923, 7828418, Mutiara Taman Palem Blok C 7 No. 20, Jl. Raya Kamal Outer Ring Road, Cengkareng Telp. 087788119800, Jl. Akses Islamic Village, Kelapa Dua, Karawaci Telp. 087788119999, Jl. Cemara Raya No. 59 Jaka Sampurna Bekasi Barat, http://www.gunadarma.ac.id, e-mail:sektor@gunadarma.ac.id

Figure A.2: Letter of data and information usage permission

MEKAR

PT Sampoerna Wirausaha

Sampoerna Strategic Square
North Tower Lt. 27
Jl. Jend. Sudirman Kav. 45-46
Jakarta Selatan 12930
Indonesia

P. 021 300 22735

SURAT KETERANGAN
No. 004/PTSW-INT.HR/I/2018

Berdasarkan surat **No. 006/Sek/Sarmag/UG/I/2018**, tentang permohonan penggunaan Data & Sistem Informasi, maka Pihak PT Sampoerna Wirausaha memberikan ijin kepada:

Nama	: **Alvian Dwi Kurnianto**
NPM	: **92216007**
Jurusan	: **Sistem Informasi Bisnis (Program SarMag)**

Untuk menggunakan Data & Sistem Informasi di PT Sampoerna Wirausaha untuk keperluan penyusunan Tesis.

Demikian surat keterangan ini kami buat dengan sebenarnya, dan untuk dipergunakan sebagaimana mestinya.

24 Januari 2018,
PT Sampoerna Wirausaha,

PT SAMPOERNA WIRAUSAHA

Dandi Rusli
CTO

www.mekar.id

CURRICULUM VITAE

PERSONAL INFORMATION

Name : Alvian Dwi Kurnianto

Birth Place and Date : Pekalongan, 4 September 1994

Sex : Male

NIM : 92216007

Address : Depok, West Java

Phone Number : 085840002746

Email : alviandk@gmail.com

FORMAL EDUCATION

2012 - 2016 : Information System Bachelor Degree at Gunadarma University

WORK EXPERIENCE

2016 - now : Mekar, Sampoerna Wirausaha - Remote Django Developer

2015 - 2017 : Coassets - Remote Django Developer

2014 - 2015 : Macroad - Django Developer Internship

2014 - 2015 : Coding Indonesia - Programming Tutor

SCIENCE PUBLICATION (2016)

2016 : Undergraduate Thesis titled “Animetacchi Website Development with Django and Ajax”